# DESCRIPTION

# METHOD OF CONTENT ADAPTATION FOR APPARATUSES OF UNIVERSAL MULTIMEDIA FRAMEWORK TERMINAL, SERVER, AND GATEWAY

## TECHNICAL FIELD

The present invention relates to a unified multimedia terminal that can access multimedia content in different formats, and relates to a unified multimedia framework that can handle different multimedia terminal, network, usage environment.

## BACKGROUND ART

MPEG and other standard bodies have created many standards in video, audio, system, communication protocol, content representation, content packaging, etc., to ease for content transfer and deliver from one place to another in an efficient way, as well as content storage with big volume in a limited space.

As a result of it existing contents can be found in different formats due to the different source coding methods, MPEG1, MPEG2, MPEG4, DV, DVCPRO, etc, and at the same time devices and terminals are also built in different scope to be compliant with certain standard but not with another. For example, a DTV STB (Set-Top-Box) can decode and play back MPEG-2 compliant content but cannot play back DV content, or MPEG-4 content.

It would be desirable that there has a universal multimedia terminal, which can access and consume any content no matter what format it is. There are a few ways to achieve this.

One way is to build a terminal that has a powerful transcoder function to understand all the different formats and decode them in real-time, then encode the content into the required format in real-time. But the cost would be very high for having such a real-time multifunctional transcoder. It is definitely not advisable to use for portable terminal, even for DTV, or DVD like devices.

Another way is to have such transcoder function in service provider side / content creation side, or in a gateway site, to pre-transcode and store the same content in different formats.

In such case a set of descriptions or metadata to describe content representation format can be created and attached to the corresponding content. By matching a query submitted by a terminal with the stored metadata in a server, the server will decide which content in the specific format will be delivered to the terminal. Here the terminal has to be a compliant terminal, to be able to generate a set of pre-defined descriptions or metadata that the server can recognize.

Besides the set of descriptions to address content representation format, a set of descriptions to describe terminal, as well as set of descriptions to describe other usage environment including network condition and user preference, is also required to define. However usage environment is different from content representation format and terminal, and it is changeable and selectable and not fixed with the terminal like terminal decoding capability.

Here this invention is focusing on defining a set of descriptions and the structure among the descriptions, as well as to define a mechanism for a multimedia framework used across different terminals and network.

This invention is to try to solve the following problems:

The same content or media resource in certain format can be accessed, retrieved, and consumed by any multimedia terminal with different network condition and different user preference.

A terminal with different decoding capability and network condition is able to access, retrieve, and consume any content or media resource that is in different data formats.

DISCLOSURE OF INVENTION

By defining a set of descriptions for content representation format, a piece of content or media resource can be described to reflect its coding format and

style, where the set of descriptions can be defined based on coding condition according to certain coding standard, and also based on parameters extracted from the bit stream headers.

By defining a set of descriptions for a terminal, the terminal can be described to reflect its conformed decoding capability.

By defining a set of descriptions for a terminal, the terminal can be described to reflect its processing power, memory, operating system, etc. Such description is sent to a server from the terminal before the server can deliver content in the right coding format.

By defining a set of descriptions for network, the network condition which a terminal is connected to, can be described to reflect communication protocol, average bandwidth, delay status, packet loss status, etc.

By defining a set of descriptions for user preference, user's interest and preference can be described to reflect user's choice and selection.

By defining the hierarchy and structure of the above descriptions, a multimedia framework can be built for content delivery to any terminal via any network.

By defining the hierarchy and structure of the above descriptions, a terminal can be built to form a universal multimedia terminal that may have limited decoding capability.

By defining a mechanism used for constructing a multimedia framework, many of the existing and future applications can be served for more users who have different terminals and different network.

Operation of the Invention

On the terminal side:

A MPEG-21 Terminal is built by implementing one of the source coding tools like MPEG-4 Video decoder for simple profile;

A MPEG-21 Terminal is built by implementing one of the system coding tools like MP4 file format;

A MPEG-21 Terminal is built by implementing one of the Transmission tools like RTP;

A MPEG-21 Terminal is built by implementing MPEG-21 DID (Digital Item Declaration), REL (Rights Expression Language), RDD (Rights Data Description), IPMP (Intellectual Property Management and Protection), and DIA (Digital Item Adaptation) Parser;

A MPEG-21 Terminal is built by setting-up a table to describe the terminal's characteristics such as compliant decoding format including source coding format, system layer coding format, transmission format, where the table can be expressed in XML (Extensible Markup Language 1.0) schema or in SDL (Syntactic Description Language) syntax, where more detail descriptions will be included in each type of coding format, such as bit rate, display window size, buffer size, etc.

On the content server side:

A content or media resource that is in certain data format is pre-transcoded into different data formats with its defined metadata or a set of defined DIA descriptions to describe each type of formats according to MPEG-21 DIA defined structure.

MPEG-21 Digital Item (DI) is generated by constructing a scene structure where the corresponding DID/IPMP/REL/RDD/DIA descriptions are structurally linked together to tell you what is the digital item and where is the media resource, how it is protected, what are the rights for using the DI, and what is the possible terminal and network adaptation listed in form of "choice" for the media resource in order to be delivered and consumed by users. This is used for two-way interactive application.

For one-way application like broadcasting, MPEG-21 Digital Item will be created according to the specific terminal and network condition received from terminal.

It should be noted that the terms and definitions given in the following apply:

Digital Item: a Digital Item is a structured digital object with a standard representation, identification and metadata within the ISO/IEC 21000 framework. This entity is also the fundamental unit of distribution and transaction within this framework

Digital Item Adaptation: a process where a Digital Item is subject to a resource adaptation engine and/or a descriptor adaptation engine, which together produce an adapted Digital Item.

Content Digital Item: a Digital Item that is used for the delivery of resources.

## BRIEF DESCRIPTION OF DRAWINGS

The present invention will become readily understood from the following description of preferred embodiments thereof made with reference to the accompanying drawings, in which like parts are designated by like reference numeral, and in which:

Fig. 1 shows the Prior Art 1- One-way Content Delivery to Users with Different Types of Terminals;

Fig. 2 shows Prior Art 2 - Two-way Content Retrieval and Accessing by Users with Different Types of Terminals;

Fig. 3 shows A Terminal with MPEG-21 DIA Descriptions and MPEG-21 DIA Parser;

Fig. 4 shows Hierarchy DIA Descriptions for Each Layer to Describe a Media Resource;

Fig. 5 shows an Architecture of Terminal Description XML Schema;

Fig. 6 shows the Architecture of Terminal Description XML Schema continued from Fig. 5;

Fig. 7 shows DIA Descriptions for a Media Resource and a Terminal;

Fig. 8 shows Illustration of an Adaptation Gateway from one format with its DIA Descriptions to another format with its DIA descriptions;

Fig. 9 shows Illustration of MPEG-21 Terminal with DIA Description used in

one-way Application;

Fig. 10 shows Illustration of MPEG-21 Terminal with DIA Description used in Interactive two-way Application;

Fig. 11 shows Real-Time Adaptation Framework for Streaming Content Delivery; and

Fig. 12 shows a Generic Adaptation Framework to a Terminal with Different Network Condition and User Preference.

## BEST MODE FOR CARRING OUT THE INVENTION

The prior art is illustrated in Fig. 1 and Fig. 2 to state the current situation where a content in certain data format cannot be played back by different terminals.

In Fig. 1, it is shown that MPEG-2 MP@ML content is delivered from Multimedia server in module 1.1 to user in one-way, and only DTV STB in module 1.4 can play back. The other terminals in module 1.2, 1.3, and 1.5 cannot play the content.

In Fig. 2, it is shown that MPEG-2 MP@ML content is delivered from Multimedia server in module 2.1 to user in interactive two-way, and only DTV STB in module 2.4 can play back. The other terminals in module 2.2, 2.3, and 2.5 cannot play the content.

From the above examples it is understood that such situation limits the content accessibility by a broad range of terminals and network.

On the other hand, it is not possible for terminal to be built to enable decoding all different formats of content, which is too much costly.

The following embodiments are arranged in the two major sections:

Hierarchy DIA Descriptions for adaptation to different terminals used in Interactive Two-way or One-way applications and real time streaming adaptation to different network;

Interactive Two-way application case

The interactive application is very popular in existing world, such as DTV Interactive data broadcasting, Internet based services, Interactive DVD, Interactive data access by Mobile phone, etc.

Typical model for this two-way application is that server provides a layer structured menu to be viewed by User, User selects what he want to retrieve by sending request to server from a client terminal, and the server processes the request and delivers the requested content to the client terminal.

Here the content is referred to media resource in MPEG-21.

As shown in Fig. 3, MPEG-21 Multimedia Server is illustrated in module 3.1 where content or media resource is stored attached with a set of DIA descriptions.

In module 3.11 a MPEG-21 terminal is shown and it consists of source coding tool in module 3.10, system layer coding tool in module 3.9, transmission protocol in module 3.8, as well as MPEG-21 DIA parser and a set of DIA descriptions in module 3.7. The terminal is connected to TCP/IP, UDP, ATM, or other network in module 3.6.

The case what we are talking here is corresponding to Case 2: Two-way Interactive Application shown in Fig. 3. First of all, the terminal or user will browse a Digital Item, parse the DID/IPMP/REL/RDD and DIA descriptions based on what a user selects in module 3.5. The server delivers the content with matched data format to the terminal according to the choices made by the terminal, in module 3.4.

In the above interactive process, the information received by the server consists of two types: User involved and Non-user involved.

1) User involved information: those requests or choices made by the user during the browsing, such as user preference;

2) Non-user involved information: the terminal characteristics sent from the terminal automatically when the DIA descriptions are parsed through the DIA Parser in the terminal.

One-way Broadcasting Application Case

In this way, terminal is not supposed to communicate with the server interactively like two-way application.

As shown in Fig. 3 for Case 1: one-way Application, the terminal needs to initiate the content delivery by sending a request with a set of DIA descriptions, to tell the server "what I am" and "what I can do" in module 3.2. The server processes the request and the set of DIA descriptions to transmit the content with the matched format to the terminal, in module 3.3.

There are two types of DIA descriptions. One type is related to terminal characteristics, such as terminal complaint decoding format, terminal process power, etc., and this type of the descriptions is fixed with the terminal and not likely be selected by User; while the other type is related to network condition and user preference which is not fixed with the terminal and it is changeable with different network condition and different user preference.

No matter one-way or two-way applications, those DIA descriptions related to terminal characteristics have to send to the server by terminal not by user. For one-way application they can be submitted to server during communication starting period, while for two-way application they can be submitted to server in the beginning or in the middle way of interactive communication. While those DIA descriptions related to network condition and user preference can be given by user or network protocol that is used in the communication during the delivering process.

Hierarchy Description for Each Coding Layer:

As shown in Fig. 4, a general media resource in module 4.1 can be in the form of text, graphics, still image, audio, video, which are considered as source coding format in module 4.2.

If there is combination of audio and video, system coding is required as shown in module 4.3 to synchronize audio and video, such as MPEG-2 Transport system and Program system which are widely used in the products.

In some case a transmission protocol is required as shown in module 4.4, to achieve certain transmission purpose, such as real time transmission.

Such a media resource is formed using various types of coding format, and a set of descriptions is generated to tell the coding structure of the media resource, as shown in module 4.5, called MPEG-21 DIA Description.

Such descriptions and the layer structure among different layers can be expressed in XML or in MPEG-21 file format in the form of SDL (Syntactic Description Language). Here a XML based schema is used to describe such DIA description as one of the possible solutions (the structure of schema is shown in Fig. 5 and Fig. 6 and the detail schema can be found in the end of this section).

In Fig. 5 and Fig. 6 schema structure, we define an exhaustive terminal description including "General" description (5.1), "Hardware" (5.2), "Software" (5.3), and "System" (5.4) properties. The hierarchy description of terminal coding capability can be seen in "CodingCapability" element (5.5).

In general terminal description, the terminal vendor, model and its type/class such as encoder, decoder, gateway, router, PC, PDA, printer, Mobilephone etc are described.

The hardware and software components are two important elements to denote the terminal capability. The brief description of each element under these two parts are listed below:

CPU: Vendor, Model, Speed of the device CPU, e.g. Intel Pentium III 1.13GHz;

Memory: Vendor, Model, Size of the device memory, e.g. Intel SDRAM PC133 128M;

Speaker: Vendor, Type of the device speaker, e.g. Yamaha YMF753;

Keyboard: Vendor, Type of the device keyboard, e.g. Toshiba 85-key Ergonomic Keyboard;

Screen: it gives more detail of terminal screen,

"Size": device screen size in unit of pixels, composed of the horizontal and vertical values; "SizeChar": device screen size in units of characters, composed of the horizontal and vertical values and determined by device's standard font; "BitPerPixel": the number of bits of colour or grayscale information per pixel, related to the number of colours or shades of gray the device can display in screen; "PointingResolution": the type of resolution of the pointing accessory supported by device screen.

Hardware Support: indicate whether the terminal supports "Colour" (colour display), "TextInput" (text entry), "ImageInput" (image display), "VoiceInput" (any form of voice input), "SoftKey" (Programme soft keys) and "SoundOutput" (sound output through external speaker); Boolean type is used to denote all these supports, e.g. "yes" for "Colour" support means the device's display support colour";

AssitHardware: some assistant hardware supporting security functionality, "SmartCard", "Hardkey";

OS: Vendor, Name, Version, SerialNumber of the terminal's operating system, e.g. Apple Mac OS, 9.0, 12345678;

VM: Vendor, Name, Version of the virtual machine installed on the device, e.g. SunJRE1.2;

Firmware: Vendor, Name, Version of the firmware to which the device's low-level software conforms;

Software Support: indicate whether the terminal supports "Software download", "Browser" with its name and version;

System property is also an important aspect for terminal description.

General description of terminal system including processing "Modules", e.g. extra terminal tools for some special functionalities' performing, interconnection of "Components" and the support of "Configuration" option. For MPEG-21 multimedia framework, the special system property "IPMP" should be provided. The IPMP system description includes residing IPMP_Tools with their ToolID

and the support of RDD/REL.

The terminal coding capability is described in a structured mode according to content type (Audio, Video, Image/Graphic, Text) and the respective coding type (e.g. MPEG, ITU). In each coding type, profiles and levels are used for further classifying and extending. Besides these coding formats description, the coding parameters ("ControlTerms") such as Picture Size, Display Widow Size, Bit Rate, Frame Rate, Buffer Size, etc are the possible control items to use to describe coding format. The details can be found in the following video coding structuring explanation and the terminal schema description.

The detail structuring for video is also shown as in Fig. 4 to illustrate what are the possible items or parameters required to describe video.

As for MPEG-n video coding format, there are MPEG1 video, MPEG2 video, MPEG4 video, and also MPEG4 AVC (Advance Video Coding). Under each MPEG video coding format, there is profile, level, version to tell the compliant level.

So for a given media resource, what profile, what level, and what version has to be given to just describe its video coding format. It is shown as in module 4.6, 4.7, and 4.8 in Fig. 4, respectively, where Simple profile with Level 1 and Version 1 is indicated for the video coding.

On top of that, coding parameters are also required to be included in the DIA video coding descriptions, which is shown in module 4.9, where Picture Size: W/H (Width and Height), Display Widow Size: W/H, Bit Rate, Frame Rate, Buffer Size, and others if there are, are the possible items to use to describe video coding format.

Here video can include primary video in different video formats, like CCIR601 4:2:2, RGB with 8 bits for each colour component, etc..

Video format also covers DVxxx that includes DV (Digital Video) format used in DV camera, DVCPRO-25, DVCPRO-50, DVCPRO-HD. These are the popular format found in the original content and captured by digital camera.

H.26x consists of H.261 and H.263, and mainly used in videoconference products;

M-JPEG is motion JPEG;

WMV is Window Media Video used widely in PC world;

Real Video is another type of video format used in the product of Real Network;

In the same way as video, Audio and speech also need to cover different formats: MPEG-n audio like MPEG Layer 2, MP3, MPEG2-AAC, MPEG4-AAC, and G series like G.721/2, G.723, G.726, G.729, etc. Image and graphics covers various types of formats, like JPEG, JPEG-2000, GIF, TIFF, XBM, PICT, etc.

DIA description to describe a content structure and a terminal is shown in Fig. 7.

In module 6.1, DIA descriptions are listed to describe a media resource, while in module 6.2, DIA descriptions are listed to describe a multimedia terminal.

In the front of a content structure, a set of DIA descriptions is always attached by described the detail coding structure of the content, as shown in module 6.3.

On the other hand, for a given terminal a set of DIA descriptions is always requested to be implemented with the terminal to describe the terminal decoding capability and its processing power, as shown in 6.4.

From Fig. 7, it is clear that two formats in module 6.1 and 6.2 do not match and adaptation is required to enable the same content for the media resource can be consumed by the terminal.

The ideal adaptation is to rely on a real-time Adaptation Gateway that contains transcoding module. Once the adaptation gateway receives such DIA descriptions as listed in module 6.1, it will convert or transcode the coding format to fit for the DIA description as listed in module 6.2 for consuming

terminal. In this case, for Source Coding Layer, MPEG-2 AAC is transcoded into MPEG-4 AAC, and MPEG-2 video MP@ML (Main Profile and Main Level) transcoded into MPEG-4 video Simple Profile with Level 1 and Version 1. For System Coding Layer, MPEG-2 Transport System is converted and transcoded into MP4 file format. A downsize process is required here before encoding into MPEG-4 Video format from MPEG-2 video due to the different supported Picture Sizes on two sides indicated in Coding Parameters in the two sets of DIA descriptions.

The adaptation gateway is shown in Fig. 8 to illustrate the above solution. It consists of DIA Parser and transcoding module to convert from one format into another and re-deliver to a terminal.

As shown in Fig. 8, in module 7.1 DIA Parser in an Adaptation Gateway is to process the input DIA descriptions received from a server, and in module 7.2 the processed results after module 7.1 are compared to the Terminal's DIA descriptions that are subscribed by the terminal. Based on the results in module 7.2, transcoding in Source Coding layer and System Coding layer will be performed in module 7.3, 7.4 and 7.5, respectively for audio, video, and system coding format.

Sometimes to save computation time, the conversion or transcoding can be done in bit stream level, and such intelligent trancoding tool can be even implemented in a terminal and the implementation can be different in different situations. These will be left to implementers to consider according to their terminal's process power and applications.

There is another better way to serve for the purpose. The useful working model should be relying on the processing power on server side. More important point is the application scenario where a media resource is pre-encoded and stored in a multimedia server or library. In such case, the same content can be pre-transcoded and stored in different coding formats to match with the popular and widely used cases.

A generic block diagram is shown in Fig. 9 for one of the implementations. A MPEG-21 terminal in module 8.1 with its DIA Descriptions is sending a content request to server with its set of DIA descriptions that describe the terminal complaint coding format as shown in module 6.2 in Fig. 7. The server is processing the received DIA descriptions in module 8.2, and searching for the matched coding format for the requested content in module 8.3. Finally the requested content in the matched coding format is being delivered to the terminal from the server. This case is for One-way application like broadcasting.

Another implementation is shown in Fig. 10, which is for interactive application. A terminal in module 9.1 browses a Digital Item that is described with DID/REL/RDD/IPMP/DIA in (1), and the server provides different "choices" to indicate terminal characteristics, network condition, or user preference in (2).

There are two ways for server to acquire about terminal characteristics: once user selects "Terminal Characteristics" displayed to the user in (3) as shown in Fig. 10, the terminal will automatically send its DIA Descriptions to the server, and another way is to let user decide for the terminal characteristics in the case the user knows about their terminal by any means. Finally the content with matched coding format as an adapted media resource is delivered to the terminal in (4).

Here Terminal Characteristics includes its complaint coding capability and also its processing power/memory/operating system, etc. While the terminal's complaint coding capability is corresponding to Content Representation Format that is used to describe a media resource. They are the same descriptions but just using different terms for different targeted objects: terminal or content/media resource.

Real time Streaming Adaptation to Different Network:

As explained in the above, there are two different types of DIA Descriptions: Fixed Descriptions to describe Terminal Characteristics which is fixed and cannot be changed, and Changeable Descriptions to describe network condition

and user preference, which is not fixed with the terminal, and it can be different as network used is different or changed, or user preference is different or changed.

The former descriptions with their hierarchy structure as well as the mechanism to signal adaptation and transcoding are shown in the above sections, and the later descriptions and the mechanism to signal between a server and a client terminal is described in this section.

It is assumed that the same coding format is used for a media resource and a terminal. Due to different network condition like different bandwidths the media resource needs to be adapted to the changeable condition by any means, including using scalable coding techniques in source coding layer and also application network layer.

There are many ways to change bit rate in real-time to adapt to changeable network. These include SNR scalable coding, adjusting bit rate controller, reducing frame rate, truncating bits for enhancement layer, employing Qof (Quality of Service), etc. All of these are considered as different kinds of adaptation tools, and they can be different in different cases.

In Fig. 11, a real-time adaptation framework is shown to illustrate the streaming case where network condition and user preference is fed back to a server all the time so that the server uses such information to determine what kind of media resource should be delivered to a terminal to match with the suitable bit rate.

As shown in Fig. 11, a MPEG-21 multimedia server is shown in module 10.1 where DIA Parser, Adaptation Tools, and Media Resources are included. A MPEG-21 terminal is shown in module 10.2 with DIA Description Generator in module 10.5 to generate DIA descriptions according to the Network Condition in module 10.3 and User Preference in module 10.4.

In module 10.6 a set of descriptions is listed as an example to illustrate what are the possible items needed to submit to the server.

A DIA Parser in module 10.7 in the server is to process the received DIA descriptions from the client terminal so as to decide to select the appropriate adaptation tool from the Adaptation Tool plug-in in module 10.8. Media Resource in module 10.9 is converted by adaptation tool, to adapt to the requested and suitable network condition and user preference that is described by DIA description. Finally the requested media resource is delivered from the server to the client terminal.

A generic adaptation framework is shown in Fig. 12 to illustrate how the adaptation is performed for a given terminal with limited decoding capability, different network condition and user preference.

MPEG-21 Multimedia Server is shown in module 11.1 with DIA Parser. Different adaptation tools are implemented or used as plug-in tool in the server side. Contents are stored in the server as media resources to be retrieved and delivered to user based on request.

MPEG-21 Client Terminal is shown in module 11.2, where DIA Descriptions to describe its decoding capability and processing power in module 11.6, as well as a DIA Description on-line generation module to generate DIA description on network condition and user preference in module 11.5 is included.

During the starting of the communication, the terminal will send the set of DIA descriptions on Terminal Capability to the server, as shown in the arrow of "a", to start the content delivery.

Network Condition in module 11.3 and User Preference in module 11.4 will feed back the actual network condition and user preference to the module 11.5 to generate DIA descriptions on-line anytime if it is requested. The terminal will send the set of DIA descriptions on Network Condition and User Preference to the server, as shown in the arrow of "b".

User Interface is shown in module 11.10 to obtain User Preference. Network Condition can be obtained from Network Protocols, or propriety method in module 11.11, and it is also possible to be given by user via User interface.

As shown in module 11.7, DIA Parser in the server side will process the received DIA descriptions from the terminal and select certain adaptation tools form the server or plug-in tools to convert and transcode the requested media resource into the matched media resource to match with the terminal capability and appropriate networking condition and user preference.

Adaptation tools are shown in module 11.8 on the server side, and it can also be implemented in a middle adaptation gateway as shown in Fig. 8. Media Resources shown in module 11.9 are stored in the server, and it can be pre-transcoded and stored in the server to target on several popular coding formats, to avoid real-time transcoding for many users at the same time.

The detail Terminal Description schema is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.3 U (http://www.xmlspy.com) by Huang Zhongyang
(Panasonic Singapore Laboratories Pte Ltd) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="TERMINAL">
    <xs:annotation>
      <xs:documentation>It give the description of terminal
capabilities</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="General" minOccurs="0">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
              <xs:element name="Device_Type" minOccurs="0">
                <xs:simpleType>
```

```
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="Decoder"/>
                        <xs:enumeration value="Encoder"/>
                        <xs:enumeration value="Gateway"/>
                        <xs:enumeration value="Router"/>
                        <xs:enumeration value="Camera"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Device_Class" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="PC"/>
                        <xs:enumeration value="PDA"/>
                        <xs:enumeration value="STB"/>
                        <xs:enumeration value="Printer"/>
                        <xs:enumeration value="MobilePhone"/>
                        <xs:enumeration value=""/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Vendor" type="xs:string"
minOccurs="0"/>
            <xs:element name="Model" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Hardware" minOccurs="0">
```

```
<xs:complexType>
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="CPU" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="Vendor" type="xs:string"
use="optional"/>
        <xs:attribute name="Model" type="xs:string"
use="required"/>
        <xs:attribute name="Speed" type="xs:integer"
use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Memory" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="Vendor" type="xs:string"
use="optional"/>
        <xs:attribute name="Model" type="xs:string"
use="required"/>
        <xs:attribute name="Size" type="xs:integer"
use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Speaker" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="Vendor" type="xs:string"
use="optional"/>
        <xs:attribute name="Type" type="xs:string"
use="required"/>
      </xs:complexType>
```

```
                    </xs:element>
                    <xs:element name="Keyboard" minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="Vendor" type="xs:string"
5   use="optional"/>
                            <xs:attribute name="Type" type="xs:string"
    use="required"/>
                        </xs:complexType>
                    </xs:element>
10                  <xs:element name="Screen" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence maxOccurs="unbounded">
                                <xs:element name="Size">
                                    <xs:complexType>
15                                      <xs:attribute name="horizontal"
    type="xs:integer" use="required"/>
                                        <xs:attribute name="vertical"
    type="xs:integer" use="required"/>
                                    </xs:complexType>
20                              </xs:element>
                                <xs:element name="SizeChar" minOccurs="0">
                                    <xs:complexType>
                                        <xs:attribute name="horizontal"
    type="xs:integer" use="required"/>
25                                      <xs:attribute name="vertical"
    type="xs:integer" use="required"/>
                                    </xs:complexType>
                                </xs:element>
                                <xs:element name="BitPerPixel"
```

```
type="xs:integer" minOccurs="0"/>
                    <xs:element name="PointingResolution"
type="xs:string" minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Support" minOccurs="0">
                <xs:complexType>
                  <xs:sequence minOccurs="0"
maxOccurs="unbounded">
                    <xs:element name="Color" minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Capability"
type="xs:boolean" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="TextInput" minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Capability"
type="xs:boolean" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ImageInput"
minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Capability"
type="xs:boolean" use="required"/>
                      </xs:complexType>
                    </xs:element>
```

```
<xs:element name="VoiceInput" minOccurs="0">
<xs:complexType>
<xs:attribute name="Capability" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="SoftKey" minOccurs="0">
<xs:complexType>
<xs:attribute name="Capability" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="SoundOutput" minOccurs="0">
<xs:complexType>
<xs:attribute name="Capability" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="AssitHardware" minOccurs="0">
<xs:complexType>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="SmartCard" minOccurs="0">
```

```
                    <xs:complexType>
                        <xs:attribute name="Vendor"
type="xs:string" use="optional"/>
                        <xs:attribute name="Model"
type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="HardKey" minOccurs="0">
                    <xs:complexType>
                        <xs:attribute name="type"
type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Software" minOccurs="0">
    <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="OS" minOccurs="0">
                <xs:complexType>
                    <xs:attribute name="Vendor" type="xs:string"
use="optional"/>
                    <xs:attribute name="Name" type="xs:string"
use="required"/>
                    <xs:attribute name="Version" type="xs:float"
```

```
use="required"/>
                    <xs:attribute                    name="SerialNo"
type="xs:nonNegativeInteger" use="optional"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="VM" minOccurs="0">
                <xs:complexType>
                    <xs:attribute    name="Vendor"    type="xs:string"
use="required"/>
                    <xs:attribute    name="Name"    type="xs:string"
use="required"/>
                    <xs:attribute    name="Version"    type="xs:string"
use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Firmware" minOccurs="0">
                <xs:complexType>
                    <xs:attribute    name="Vendor"    type="xs:string"
use="required"/>
                    <xs:attribute    name="Name"    type="xs:string"
use="required"/>
                    <xs:attribute    name="Version"    type="xs:string"
use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Support" minOccurs="0">
                <xs:complexType>
                    <xs:sequence                    minOccurs="0"
maxOccurs="unbounded">
```

```
<xs:element name="Download" minOccurs="0">
<xs:complexType>
<xs:attribute name="Capability" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="Browser" minOccurs="0">
<xs:complexType>
<xs:attribute name="Name" type="xs:string" use="required"/>
<xs:attribute name="Version" type="xs:float" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="System" minOccurs="0">
<xs:complexType>
<xs:sequence maxOccurs="unbounded">
<xs:element name="General" minOccurs="0">
<xs:complexType>
<xs:sequence minOccurs="0">
<xs:element name="Modules" minOccurs="0">
<xs:complexType>
```

```
                    <xs:attribute name="ModuleID"
type="xs:unsignedInt" use="required"/>
                    <xs:attribute name="Processing"
type="xs:boolean" use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Components"
minOccurs="0">
                <xs:complexType>
                    <xs:attribute name="ComponentID"
type="xs:unsignedInt" use="required"/>
                    <xs:attribute name="Interconnect"
type="xs:boolean" use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Configration"
minOccurs="0">
                <xs:complexType>
                    <xs:attribute name="Option"
type="xs:boolean" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="IPMP" minOccurs="0">
    <xs:complexType>
        <xs:sequence minOccurs="0">
            <xs:element name="ResidingTools"
```

```
minOccurs="0" maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:attribute name="ToolID"
type="xs:unsignedInt"/>
                                    </xs:complexType>
                                </xs:element>
                                <xs:element name="RDDREL" minOccurs="0">
                                    <xs:complexType>
                                        <xs:attribute name="Capability"
type="xs:boolean" use="required"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="CodingCapability" minOccurs="0">
            <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                    <xs:element name="Audio" minOccurs="0">
                        <xs:complexType>
                            <xs:choice>
                                <xs:element name="MPEG">
                                    <xs:complexType>
                                        <xs:sequence maxOccurs="unbounded">
                                            <xs:element name="MPEG-1"
minOccurs="0">
```

```
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Layer">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="I"/>
                                        <xs:enumeration
value="II"/>
                                        <xs:enumeration
value="III"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="MPEG-2"
minOccurs="0">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element
name="LowSamplingRateLayer">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="I"/>
```

```
                                        <xs:enumeration
value="II"/>
                                        <xs:enumeration
value="III"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="BackwardCompatibleMCLayer">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="I"/>
                                        <xs:enumeration
value="II"/>
                                        <xs:enumeration
value="III"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="AAC">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="LowComplexity"/>
                                        <xs:enumeration
value="MainProfile"/>
```

```
                                        <xs:enumeration
value="SamplingRateScaleableProfile"/>
                                        <xs:enumeration
value=""/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
                <xs:element          name="MPEG-4"
minOccurs="0">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element
name="SyntheticProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                        <xs:enumeration
value="Level3"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
```

```
name="SpeechProfile">
                                        <xs:simpleType>
                                          <xs:restriction
base="xs:string">
                                            <xs:enumeration
value="Level1"/>
                                            <xs:enumeration
value="Level2"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
                                      <xs:element
name="ScalableProfile">
                                        <xs:simpleType>
                                          <xs:restriction
base="xs:string">
                                            <xs:enumeration
value="Level1"/>
                                            <xs:enumeration
value="Level2"/>
                                            <xs:enumeration
value="Level3"/>
                                            <xs:enumeration
value="Level4"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
                                      <xs:element
name="MainProfile">
```

```
                    <xs:simpleType>
                        <xs:restriction
base="xs:string">
                            <xs:enumeration
value="Level1"/>
                            <xs:enumeration
value="Level2"/>
                            <xs:enumeration
value="Level3"/>
                            <xs:enumeration
value="Level4"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element
name="HighQulityProfile">
                    <xs:simpleType>
                        <xs:restriction
base="xs:string">
                            <xs:enumeration
value="Level1"/>
                            <xs:enumeration
value="Level2"/>
                            <xs:enumeration
value="Level3"/>
                            <xs:enumeration
value="Level4"/>
                            <xs:enumeration
value="Level5"/>
```

```
                                        <xs:enumeration
value="Level6"/>
                                        <xs:enumeration
value="Level7"/>
                                        <xs:enumeration
value="Level8"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="LowDelayProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                        <xs:enumeration
value="Level3"/>
                                        <xs:enumeration
value="Level4"/>
                                        <xs:enumeration
value="Level5"/>
                                        <xs:enumeration
value="Level6"/>
                                        <xs:enumeration
value="Level7"/>
                                        <xs:enumeration
```

```
value="Level8"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
5                                           <xs:element
name="NaturalProfile">
                                                <xs:simpleType>
                                                    <xs:restriction
base="xs:string">
10                                                      <xs:enumeration
value="Level1"/>
                                                        <xs:enumeration
value="Level2"/>
                                                        <xs:enumeration
15  value="Level3"/>
                                                        <xs:enumeration
value="Level4"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
20                                          </xs:element>
                                            <xs:element
name="MobileInternetworkingProfile">
                                                <xs:simpleType>
                                                    <xs:restriction
25  base="xs:string">
                                                        <xs:enumeration
value="Level1"/>
                                                        <xs:enumeration
value="Level2"/>
```

```
                                        <xs:enumeration
value="Level3"/>
                                        <xs:enumeration
value="Level4"/>
                                        <xs:enumeration
value="Level5"/>
                                        <xs:enumeration
value="Level6"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ITU-T">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="G.723"/>
                <xs:enumeration value="G.723.1"/>
                <xs:enumeration value="G.726"/>
                <xs:enumeration value="G.728"/>
                <xs:enumeration value="G.729"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="OtherCoding">
```

```
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="AC3"/>
                                            <xs:enumeration value="DTS"/>
                                            <xs:enumeration value="AMR"/>
                                            <xs:enumeration value="LPCM"/>
                                            <xs:enumeration value="ATRAC"/>
                                            <xs:enumeration value="ATRAC2"/>
                                            <xs:enumeration value="ATRAC3"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                            </xs:choice>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="Video" minOccurs="0">
                        <xs:complexType>
                            <xs:choice>
                                <xs:element name="MPEG">
                                    <xs:complexType>
                                        <xs:sequence maxOccurs="unbounded">
                                            <xs:element          name="MPEG-1"
type="xs:boolean" minOccurs="0"/>
                                            <xs:element          name="MPEG-2"
minOccurs="0">
                                                <xs:complexType>
                                                    <xs:choice>
                                                        <xs:element
name="SimpleProfile">
```

```
                                    <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                            <xs:enumeration
5   value="MainLevel"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element
10  name="MainProfile">
                                    <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                            <xs:enumeration
15  value="LowLevel"/>
                                            <xs:enumeration
    value="MainLevel"/>
                                            <xs:enumeration
    value="High1440Level"/>
20                                          <xs:enumeration
    value="HighLevel"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
25                              <xs:element
    name="SNRScalableProfile">
                                    <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
```

```
                value="LowLevel"/>
                value="MainLevel"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element
        name="SpatialScalableProfile">
        <xs:simpleType>
            <xs:restriction
        base="xs:string">
                value="MainLevel"/>
                value="High1440Level"/>
                value="HighLevel"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element
        name="HighProfile">
        <xs:simpleType>
            <xs:restriction
        base="xs:string">
                value="MainLevel"/>
```

```
value="High1440Level"/>
                                        <xs:enumeration
value="HighLevel"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="MultiviewProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="MainLevel"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="V422Profile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="MainLevel"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
```

40

```
<xs:element name="MPEG-4"
minOccurs="0">
<xs:complexType>
<xs:choice>
<xs:element
name="SimpleProfile">
<xs:simpleType>
<xs:restriction
base="xs:string">
<xs:enumeration
value="Level0"/>
<xs:enumeration
value="Level1"/>
<xs:enumeration
value="Level2"/>
<xs:enumeration
value="Level3"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element
name="ScalableProfile">
<xs:simpleType>
<xs:restriction
base="xs:string">
<xs:enumeration
value="Level1"/>
<xs:enumeration
value="Level2"/>
```

```
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element
name="AdvancedSimpleProfile">
                <xs:simpleType>
                    <xs:restriction
base="xs:string">
                        <xs:enumeration
value="Level0"/>
                        <xs:enumeration
value="Level1"/>
                        <xs:enumeration
value="Level2"/>
                        <xs:enumeration
value="Level3"/>
                        <xs:enumeration
value="Level4"/>
                        <xs:enumeration
value="Level5"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element
name="CoreProfile">
                <xs:simpleType>
                    <xs:restriction
base="xs:string">
                        <xs:enumeration
```

```
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="CoreScalableProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                        <xs:enumeration
value="Level3"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="AdvancedCoreProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
```

```
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element
name="MainProfile">
                <xs:simpleType>
                    <xs:restriction
base="xs:string">
                        <xs:enumeration
value="Level2"/>
                        <xs:enumeration
value="Level3"/>
                        <xs:enumeration
value="Level4"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element
name="NbitProfile">
                <xs:simpleType>
                    <xs:restriction
base="xs:string">
                        <xs:enumeration
value="Level2"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element
name="AdvancedRealTimeSimpleProfile">
```

```
                                        <xs:simpleType>
                                          <xs:restriction
    base="xs:string">
                                            <xs:enumeration
5   value="Level1"/>
                                            <xs:enumeration
    value="Level2"/>
                                            <xs:enumeration
    value="Level3"/>
10                                          <xs:enumeration
    value="Level4"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
15                                    <xs:element
    name="AdvancedCodingEfficiencyProfile">
                                        <xs:simpleType>
                                          <xs:restriction
    base="xs:string">
20                                          <xs:enumeration
    value="Level1"/>
                                            <xs:enumeration
    value="Level2"/>
                                            <xs:enumeration
25  value="Level3"/>
                                            <xs:enumeration
    value="Level4"/>
                                          </xs:restriction>
                                        </xs:simpleType>
```

```
                                    </xs:element>
                                    <xs:element
    name="SimpleStudioProfile">
                                      <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                          <xs:enumeration
    value="Level1"/>
                                          <xs:enumeration
    value="Level2"/>
                                          <xs:enumeration
    value="Level3"/>
                                          <xs:enumeration
    value="Level4"/>
                                        </xs:restriction>
                                      </xs:simpleType>
                                    </xs:element>
                                    <xs:element
    name="CoreStudioProfile">
                                      <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                          <xs:enumeration
    value="Level1"/>
                                          <xs:enumeration
    value="Level2"/>
                                          <xs:enumeration
    value="Level3"/>
                                          <xs:enumeration
```

```
value="Level4"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
5                           <xs:element
name="FGSProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
10                                      <xs:enumeration
value="Level0"/>
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
15  value="Level2"/>
                                        <xs:enumeration
value="Level3"/>
                                        <xs:enumeration
value="Level4"/>
20                                      <xs:enumeration
value="Level5"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
25                          <xs:element
name="SimpleFaceAnimationProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
```

```
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="SimpleFBAProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element
name="BasicAnimatedTextureProfile">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="Level1"/>
                                        <xs:enumeration
value="Level2"/>
                                    </xs:restriction>
```

```
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element
    name="ScalableTextureProfile">
                                    <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                            <xs:enumeration
    value="Level1"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element
    name="AdvancedScalableTextureProfile">
                                    <xs:simpleType>
                                        <xs:restriction
    base="xs:string">
                                            <xs:enumeration
    value="Level1"/>
                                            <xs:enumeration
    value="Level2"/>
                                            <xs:enumeration
    value="Level3"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element
    name="HybridProfile">
                                    <xs:simpleType>
```

```
                    <xs:restriction
base="xs:string">
                        <xs:enumeration
value="Level1"/>
                        <xs:enumeration
value="Level2"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="ITU_T">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="H261"/>
            <xs:enumeration value="H263"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="M-JPEG">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration
value="mjp2Profile"/>
            <xs:enumeration
```

```
value="mj2sProfile"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="OtherCoding">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="RealVideo"/>
                                        <xs:enumeration value="WMV"/>
                                        <xs:enumeration
value="QuickTime"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
                <xs:element name="ImageGraphics" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="JPEG"/>
                            <xs:enumeration value="JPEG2000"/>
                            <xs:enumeration value="TIFF"/>
                            <xs:enumeration value="GIF"/>
                            <xs:enumeration value="XBM"/>
                            <xs:enumeration value="PNG"/>
                            <xs:enumeration value=""/>
                        </xs:restriction>
                    </xs:simpleType>
```

```
                    </xs:element>
                    <xs:element name="Text" minOccurs="0">
                        <xs:complexType>
                            <xs:choice>
                                <xs:element name="ASCII" type="xs:boolean"/>
                                <xs:element name="Non-ASCII">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="HTML"/>
                                            <xs:enumeration value="ISO-8859-1"/>
                                            <xs:enumeration value="Shift_JIS"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                            </xs:choice>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="ControlTerm" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence maxOccurs="unbounded">
                                <xs:element name="DisplayWinSize" minOccurs="0">
                                    <xs:complexType>
                                        <xs:attribute name="Width" type="xs:integer" use="required"/>
                                        <xs:attribute name="Height" type="xs:integer" use="required"/>
                                    </xs:complexType>
```

```
                    </xs:element>
                    <xs:element name="PictureSize"
minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="Width"
type="xs:integer" use="required"/>
                            <xs:attribute name="Height"
type="xs:integer" use="required"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="Bitrate" type="xs:integer"
minOccurs="0"/>
                    <xs:element name="BufferSize"
type="xs:integer" minOccurs="0"/>
                    <xs:element name="FrameRate"
type="xs:integer" minOccurs="0"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

EFFECTS OF INVENTION

A multimedia framework built with the defined mechanism is able to adapt to

different formats of content across different terminals and networks, which will be used in content retrieval, content delivery, Video-on-Demand, digital library service, etc.

A terminal built with the defined structured DIA descriptions is able to access, retrieve, and consume content in different formats, and it is able to use in different applications and different network conditions to expand its usefulness.

An adaptation gateway built with the defined structured DIA descriptions is able to parse and process the input content with its DIA descriptions, to convert the content into another format to match with the required format used in the supported terminals.

A server built with the defined structured DIA descriptions is able to select adaptation tools to convert content from one into another format, to match with the required format by a terminal, so that it can serve for variety of terminals to expand its business scale.

A mechanism is introduced to use such DIA Descriptions in real-time streaming content delivery applications where both terminal and server are implemented with such DIA Descriptions Generation and DIA Parser.

The invention can have the following structures viewed from various aspects. According to the first, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, including following steps of:

building a terminal with content decoding tools that is compliant to certain standard;

describing the terminal using DIA (Digital Item Adaptation) Descriptions;

attaching the DIA Descriptions to the terminal;

sending and submitting the DIA Descriptions to a server for retrieving a piece of content, whereby the server is parsing and processing the DIA Descriptions;

selecting the content which is in the matched format with the DIA Description submitted by the terminal; and

delivering the content to the terminal.

According to the second, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, including the following steps of:

building a terminal with content coding tools that is compliant to certain standard;

describing the terminal using DIA (Digital Item Adaptation) Descriptions;

attaching the DIA Descriptions to the terminal;

implementing a DIA Description Generator in the terminal to be able to generate DIA Descriptions on-line;

generating DIA Descriptions on-line to describe network condition and user preference based on network protocol, other tools, or user preference in the terminal;

sending and submitting the DIA Descriptions to a server for retrieving a piece of content, whereby the server is parsing and processing the DIA Descriptions;

selecting the content which is in the matched format with the DIA Description submitted by the terminal; and

delivering the content to the terminal.

According to the third, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, including the following steps of:

building a terminal with content coding tools that is compliant to certain standard;

describing the terminal using DIA (Digital Item Adaptation) Descriptions;

attaching the DIA Descriptions to the terminal;

implementing a DIA Description Generator in the terminal to be able to generate DIA Descriptions on-line;

generating DIA Descriptions on-line to describe network condition and user preference based on network protocol, other tools, or user preference in the terminal;

implementing a DID (Digital Item Declaration) Parser, IPMP (Intellectual Property Management and Protection) Parser, REL (Rights Expression Language) Parser, RDD (Rights Data Dictionary) Parser, and DIA Parser in the terminal to be able to parse DID, IPMP, REL, RDD, and DIA descriptions

browsing a DI (digital Item) with its DID, IPMP, REL, RDD, and DIA Description;

selecting "choice" notes in DID menu by the terminal;

parsing IPMP, REL, RDD, and DIA descriptions in DID menu in the terminal if there is such description appearing in the DID menu;

submitting the DIA Descriptions used to describe the terminal, the network condition, and the user preference to a server for retrieving a piece of content when there is such DIA note met in the DID menu, whereby the server is parsing and processing the DIA Descriptions;

selecting the content which is in the matched format with the DIA Description submitted by the terminal; and

delivering the content to the terminal.

According to the fourth, method of content adaptation for an Apparatus of Universal Multimedia Framework Gateway, including the following steps of:

building an adaptation gateway with transcoding function;

building an adaptation gateway with a DIA Parser;

describing terminal using DIA (Digital Item Adaptation) Descriptions, attached to the terminal;

registering the DIA Descriptions into the adaptation gateway for each of the terminal supposed to be supported by the adaptation gateway;

receiving content with its DIA descriptions in the adaptation gateway;

parsing the received DIA descriptions used to describe the content by the DIA Parser in the adaptation gateway;

comparing the parsed DIA descriptions used to describe for the content with the DIA descriptions used to describe for the terminal;

transcoding the content if there is mismatched in term of coding formats indicated by the comparing results mentioned in the above step;

generating a new set of DIA descriptions to describe the content in the transcoded coding format done in the above step;

attaching the new set of DIA descriptions to the content in the transcoded coding format; and

sending the contents with different transcoded coding formats by the adaptation gateway to the supported terminal.

According to the fifth, method of content adaptation for an Apparatus of Universal Multimedia Framework Server, including the following steps of:

building a multimedia server with stored multimedia contents;

creating DIA descriptions for each content;

attaching the DIA descriptions to the corresponding content;

implementing a DIA Parser in the server;

implementing or plugging-in adaptation tools in the server to be able to perform transcoding functions between different multimedia formats;

receiving a set of DIA descriptions from a terminal via any means, wherein the set of DIA descriptions are used to describe for the terminal decoding capability, network condition used, as well as user preference;

parsing the set of DIA descriptions by the server;

selecting the adaptation tools to perform transcoding to convert from one format to another in the server, according to the parsing results mentioned in the above step;

generating a new set of DIA descriptions to describe the transcoded content;

attaching the new set of DIA descriptions to the transcoded content as its metadata;

delivering the content with the matched coding format, network condition, and user preference to the terminal.

According to the sixth, method of content adaptation for an Apparatus of

Universal Multimedia Framework Server, including the following steps of:

building a multimedia server with stored multimedia contents;

creating DIA descriptions for each content;

attaching the DIA descriptions to the corresponding content;

implementing a DIA Parser in the server;

implementing or plugging-in adaptation tools in the server to be able to perform transcoding functions between different multimedia formats;

pre-transcoding and storing the content in the server to several popular coding formats using the above the adaptation tools;

receiving a set of DIA descriptions from a terminal via any means, wherein the set of DIA descriptions are used to describe for the terminal decoding capability, network condition used, as well as user preference;

parsing the set of DIA descriptions by the server;

selecting the stored content in the server, which is in the required format that is described by the DIA Descriptions submitted by the terminal;

generating a new set of DIA descriptions to describe the transcoded content;

attaching the new set of DIA descriptions to the transcoded content as its metadata;

delivering the content with the matched coding format, network condition, and user preference to the terminal.

According to the seventh, method of content adaptation for an Apparatus of Universal Multimedia Framework Server, including the following steps of:

building a multimedia server with stored multimedia contents;

creating DIA descriptions for each content;

attaching the DIA descriptions to the corresponding content;

implementing a DIA Parser in the server;

implementing or plugging-in adaptation tools in the server to be able to perform transcoding functions between different multimedia formats;

providing IPMP, REL, RDD, and DIA descriptions in DID menu for user to

view and select;

making choices from the DID menu by the user who is using a terminal;

receiving a set of DIA descriptions from the terminal during the choice making, wherein the set of DIA descriptions are used to describe the terminal decoding capability;

receiving a set of DIA descriptions from the terminal during the choice making, wherein the set of DIA descriptions are used to describe the network condition and user preference;

parsing the above the set of DIA descriptions by the server;

selecting the adaptation tools to perform transcoding to convert from one format to another in the server, according to the parsing results mentioned in the above step;

generating a new set of DIA descriptions to describe the transcoded content;

attaching the new set of DIA descriptions to the transcoded content as its metadata;

delivering the content with the matched coding format, network condition, and user preference to the terminal.

According to the eighth, method of content adaptation for an Apparatus of Universal Multimedia Framework Server, including the following steps of:

building a multimedia server with stored multimedia contents;

creating DIA descriptions for each content;

attaching the DIA descriptions to the corresponding content;

implementing a DIA Parser in the server;

implementing or plugging-in adaptation tools in the server to be able to perform transcoding functions between different multimedia formats;

pre-transcoding and storing the content in the server to several popular coding formats using the above the adaptation tools;

providing IPMP, REL, RDD, and DIA descriptions in DID menu for user to view and select;

making choices from the DID menu by the user who is using a terminal;

receiving a set of DIA descriptions from the terminal during the choice making, wherein the set of DIA descriptions are used to describe the terminal decoding capability;

receiving a set of DIA descriptions from the terminal during the choice making, wherein the set of DIA descriptions are used to describe the network condition and user preference;

parsing the above the set of DIA descriptions by the server;

selecting the stored content in the server, which is in the required format that is described by the DIA Descriptions submitted by the terminal;

generating a new set of DIA descriptions to describe the transcoded content;

attaching the new set of DIA descriptions to the transcoded content as its metadata;

delivering the content with the matched coding format, network condition, and user preference to the terminal.

According to the ninth, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, according to the above, further including the following steps of:

describing the terminal by using a hierarchy structure, including terminal general feature, terminal hardware, terminal software, terminal system, and terminal decoding capability as the 1st layer;

describing the terminal decoding capability in different coding layers, from primary content, source decoding layer, system decoding layer, transmission layer, to the DIA descriptions;

defining detail descriptions in each coding layer including coding and decoding parameters, to indicate the maximum supporting ranges by the terminal for different parameters used in the decoding process;

defining controlling terms to use in the DIA descriptions mentioned in the above, which is a set of common terms and used to signal between the terminal

and the server.

According to the tenth, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, according to the above, further including the following steps of:

describing the terminal by using a hierarchy structure, including terminal general feature, terminal hardware, terminal software, terminal system, and terminal decoding capability as the 1st layer;

describing the terminal decoding capability in different coding layers, from primary content, source decoding layer, system decoding layer, transmission layer, to the DIA descriptions;

defining detail descriptions in each coding layer including coding and decoding parameters, to indicate the maximum supporting ranges by the terminal for different parameters used in the decoding process;

defining controlling terms to use in the DIA descriptions mentioned in the above, which is a set of common terms and used to signal between the terminal and the server; and

expressing the DIA descriptions in XML to provide extensibility.

According to the eleventh, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, according to the above, further including the following steps of:

describing the terminal by using a hierarchy structure, including terminal general feature, terminal hardware, terminal software, terminal system, and terminal decoding capability as the 1st layer;

describing the terminal decoding capability in different coding layers, from primary content, source decoding layer, system decoding layer, transmission layer, to the DIA descriptions;

defining detail descriptions in each coding layer including coding and decoding parameters, to indicate the maximum supporting ranges by the terminal for different parameters used in the decoding process;

defining controlling terms to use in the DIA descriptions mentioned in the above, which is a set of common terms and used to signal between the terminal and the server; and

creating the DIA descriptions in the form of SDL (Syntactic Description Language).

According to the twelfth, method of content adaptation for an Apparatus of Universal Multimedia Framework Terminal, according to the above method, wherein network condition that is related to content format adaptation is described by defining controlling terms, like bandwidth, delay, and packet loss; and

user preference that is related to content format adaptation is described by defining controlling terms, like quality, color, size, streaming, downloading, filtering with their attributes; further including the following steps of:

receiving network condition in the terminal from network protocol, other private tools, or user input;

converting the received parameters on network condition in the terminal into the DIA descriptions by using the controlling terms defined in the above steps;

receiving user preference in the terminal from User Interface; and

converting the received parameters on user preference in the terminal into the DIA descriptions by using the controlling terms defined in the above steps.

According to the thirteenth, an apparatus of universal multimedia framework terminal, the terminal connected to a server via a network, including:

a memory device in which content decoding tools are stored;

a DIA descriptor for describing the terminal using DIA description;

an attaching element for attaching the DIA descriptions to the terminal;

a transmitter for sending the DIA descriptions and a content requirement to a server for retrieving a piece of content; and

a receiver for receiving the content selected and delivered from the server.

According to the fourteenth, an apparatus of universal multimedia framework

gateway, the gateway provided between a terminal and a server, including:

a memory device;

a DIA descriptor for describing the terminal using DIA description;

a first attaching element for attaching the DIA descriptions to the terminal;

a register for registering a DIA descriptions attached to the terminal into the memory device;

a receiver for receiving a content with its DIA descriptions according to a content requirement of the terminal from the server;

a DIA parser for parsing the received DIA descriptions attached to the content;

a comparator for comparing the parsed DIA descriptions with the DIA descriptions for the terminal;

a transcoder for transcoding the content from one format to another format, if format mismatch is found between the one format of the content and another format described in the DIA descriptions for the terminal;

a generator for generating a new set of DIA descriptions to describe the transcoded content;

a second attaching element for attaching the new set of DIA descriptions to the transcoded content; and

a transmitter for sending the content to the terminal.

According to the fifteenth, an apparatus of universal multimedia framework server, the server connected to a terminal via a network, including:

a memory device in which multimedia contents are stored;

a creator for creating DIA description for the each content;

a first attaching element for attaching the DIA description to the corresponding content;

a receiver for receiving a set of DIA descriptions from a terminal, the set of DIA descriptions being used to describe for the terminal decoding capability, network condition, as well as user preference;

a DIA parser for parsing the received DIA description for the terminal;

adaptation tools for transcoding the content between different multimedia formats of contents;

a selector for selecting one adaptation tool to transcode to convert the content from one format to another format according to the parsing results;

a generator for generating a new set of DIA descriptions to describe the transcoded content;

a second attaching element for attaching the new set of DIA descriptions to the transcoded content as its metadata; and

a delivering element for delivering the content to the terminal.

Although the present invention has been described in connection with the preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications are apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims, unless they depart therefrom.